



SPS

Správa síťových prvků aplikace





- OSS

- nagios
- opennms
- zabbix
- zenoss
- cacti
- Icinga
- Prometheus
-

- komerční

- HP NNM
- HP OVO
- IBM Tivoli
- MSC operations
- SunNet Manager
-



- oznámení události
- získání informací o události
- řešení problému, eskalace
- dokumentace události



- agenti
 - získávání informací
 - provádění operací
 - automaticky
 - manuálně
- protokoly
 - standardní
 - SNMP, DMI, CMIP
 - proprietární
 - RPC, HTTPS, XML ...



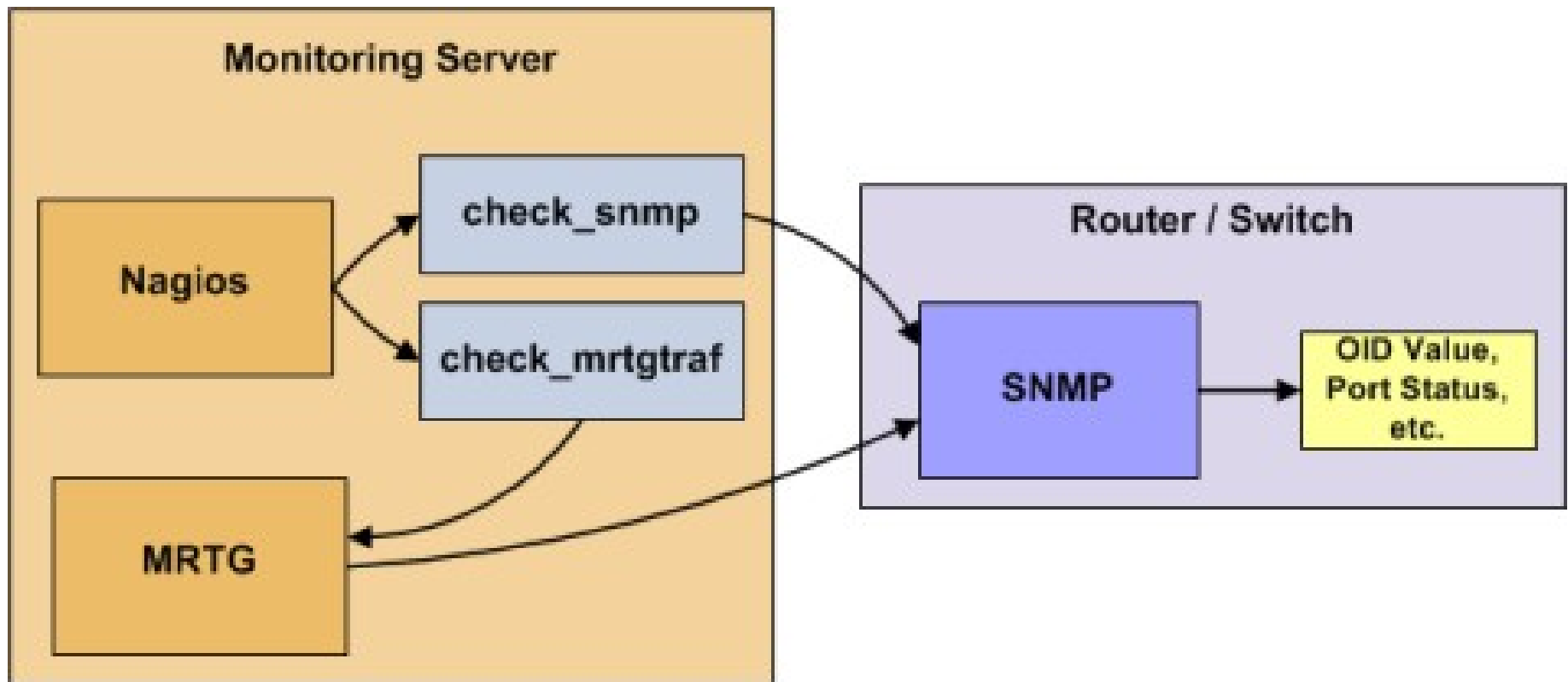
- sběr dat od agentů
- zobrazení
- správa poplachů
 - přivlastnění
 - potvrzení
 - předání
 - expertní systém
 - sledování zpracování poplachu
- distribuované zpracování
- správa služeb



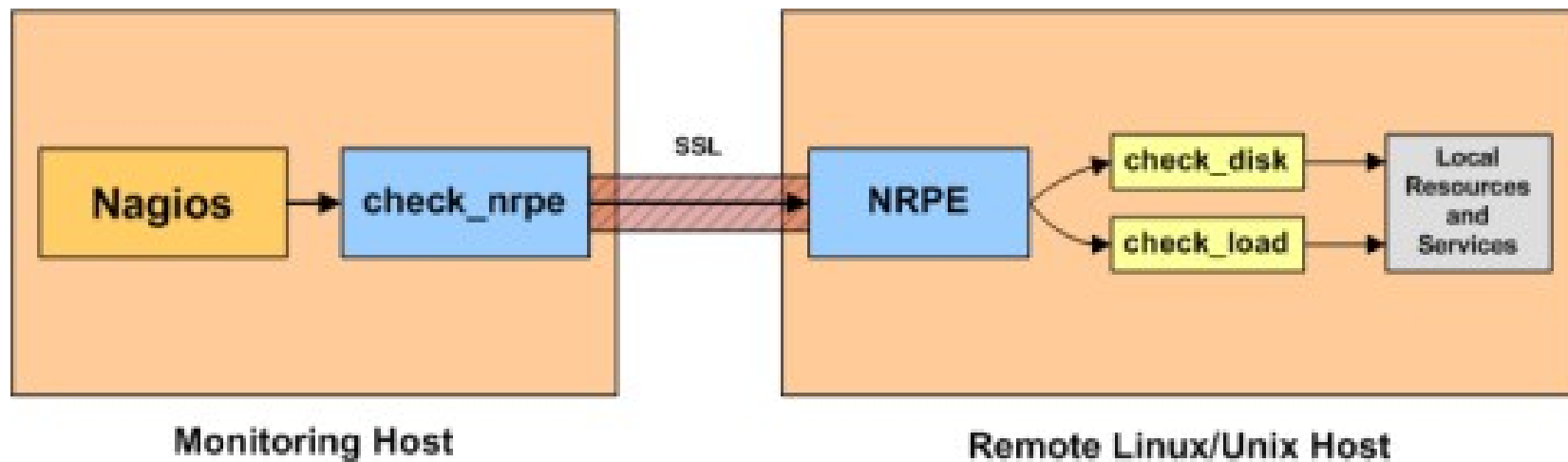
- podle atributů
 - uzel
 - text
 - důležitost
 - čas
- podle souvislostí
 - link up/down
 - výpadek směrovače a ostatních uzlů
 - korelace



- www.nagios.org
- Netsaint do 2002
- fork Icinga do 5/2009
- monitorování
 - síťové služby
 - ping, http, smtp, ...
- monitorování
 - zdroje
 - CPU, RAM, disk, ...
 - pluginy
 - oznamování
 - email, ...
 - ...

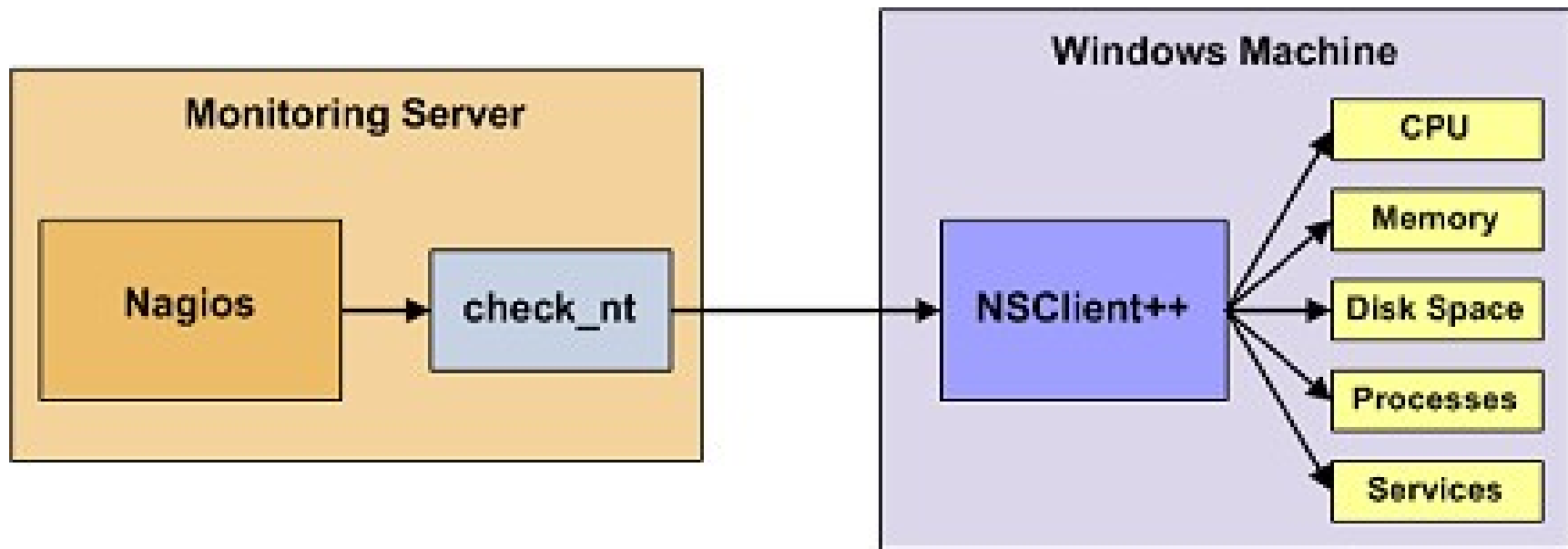


obrázky převzaty z: <http://nagios.sourceforge.net/>





MS Windows





```
define host{
    use                generic-host            ; Inherit default values from a template
    host_name          remotehost             ; The name we're giving to this host
    alias              Some Remote Host       ; A longer name associated with the host
    address            192.168.1.50          ; IP address of the host
    hostgroups         allhosts               ; Host groups this host is associated with
}

define command{
    name               check_http
    command_name       check_http
    command_line       $USER1$/check_http -I $HOSTADDRESS$ $ARG1$
}

define service{
    use                generic-service         ; Inherit default values from a template
    host_name          remotehost
    service_description HTTP
    check_command      check_http
}

define service{
    use                generic-service         ; Inherit default values from a template
    host_name          remotehost
    service_description Product Download Link
    check_command      check_http!-u /download/index.php -t 5 -s "latest-version.tar.gz"
}
```



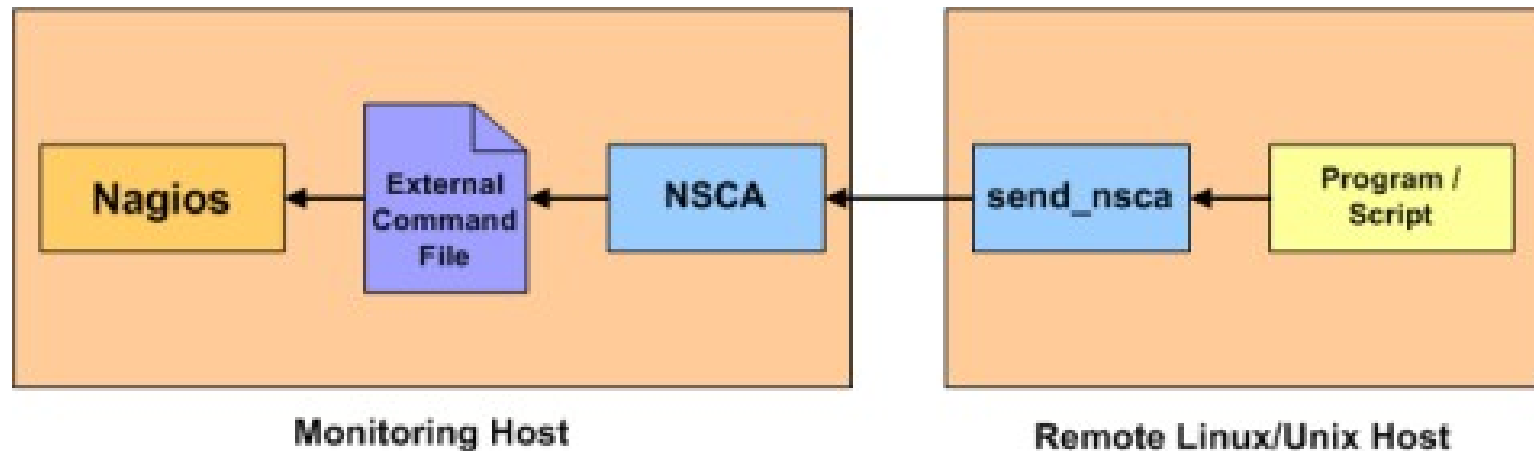
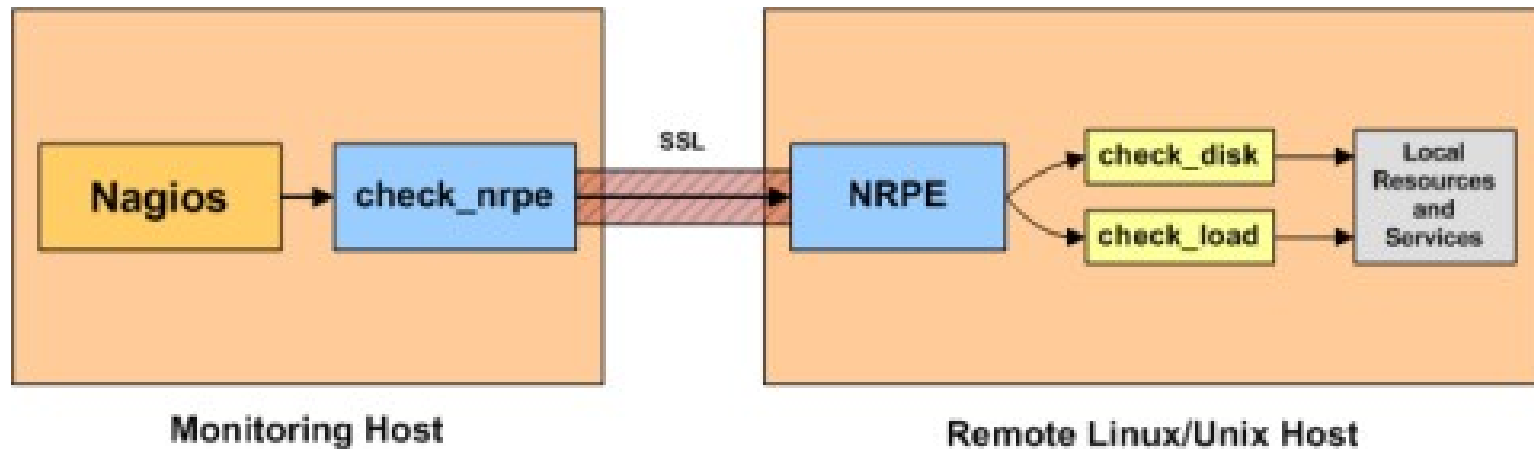
Nagios addon

approved by
dsn.felk.cvut.cz

- <http://exchange.nagios.org/>

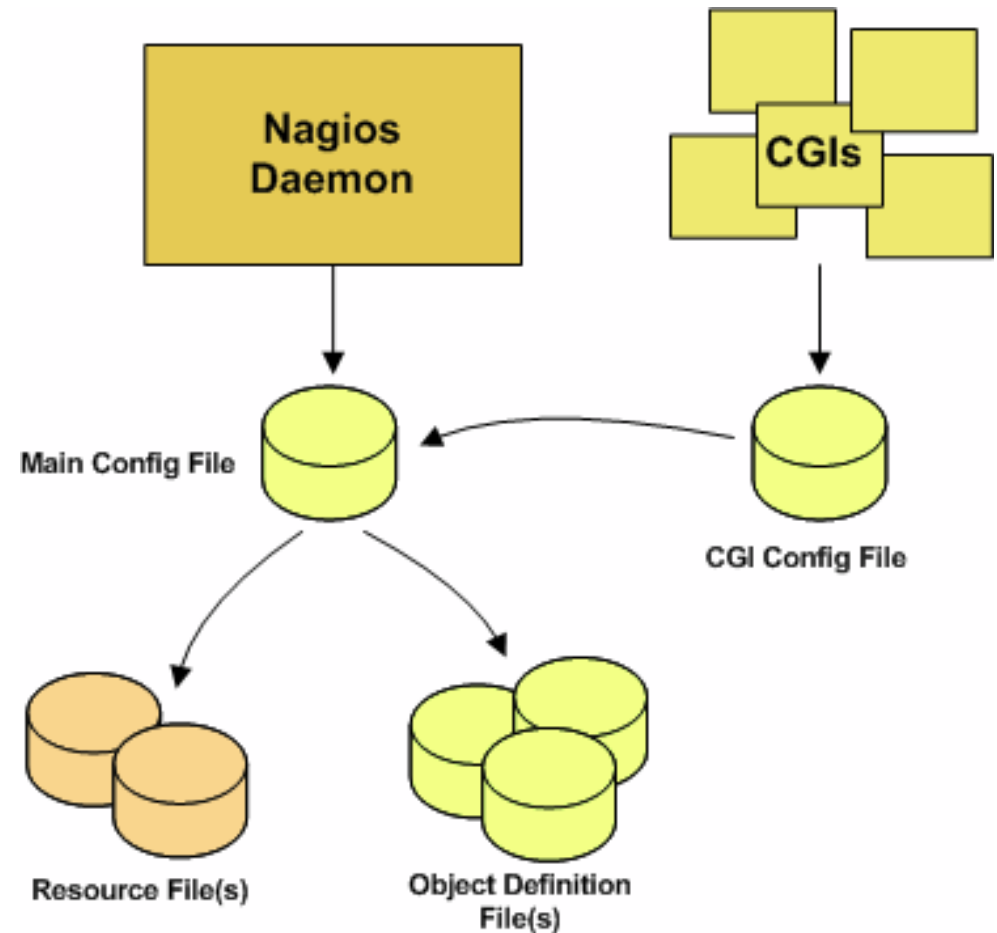


NRPE x NSCA





- /usr/local/nagios/etc
- web konfigurace
- restart nagiosu



```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```



- Services
- Service Groups
- Hosts
- Host Groups
- Contacts
- Contact Groups
- Commands
- Time Periods
- Notification Escalations
- Notification and Execution Dependencies

cfg_file=<file_name>
cfg_dir=<directory_name>



- <http://www.ibm.com/tivoli>
- https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-10_4000_100
- <http://www.nagios.org/>
- <http://www.opennms.org/>
- <http://www.zabbix.com/>
- <http://www.microsoft.com/systemcenter/>

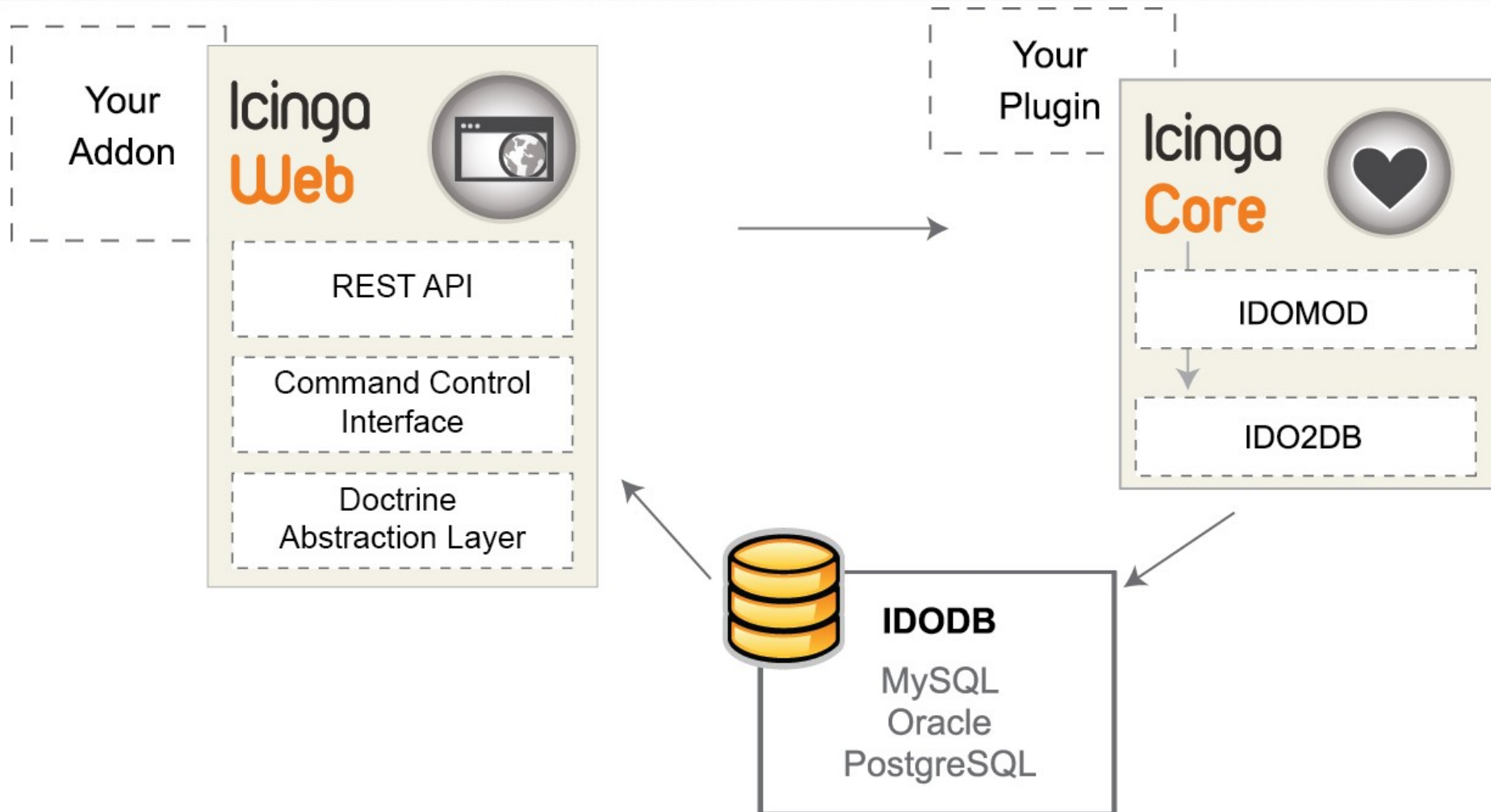


- <https://icinga.com/>
- Icinga 1
 - Binárně kompatibilní s Nagiosem
- Icinga 2
 - 2012/2014 (preview/stable)



Icinga architektura

approved by
dsn.felk.cvut.cz



Icinga's modular architecture with Core, Web and Icinga Data Out Database, as of v.1.5.
The Icinga Project, CC-BY-SA-3.0



- Zdroje ukázek
 - <https://icinga.com/docs/icinga2/latest/>
 - <https://kifarunix.com/how-to-monitor-hosts-and-services-with-icinga-2-on-ubuntu-18-04/>
- Simple Host monitoring
 - host
 - služby
 - ssh, https, ...
- Monitoring Agents
 - vzdálené služby nedostupné síťově



Simple Host monitoring

- host
 - 0,1 – UP
 - 2,3 – DOWN
 - hostalive
 - ping
 - service
 - dummy
- service
 - 0 – OK
 - 1 – WARNING
 - 2 – CRITICAL
 - 3 – UNKNOWN
 - check plugins
 - HARD
 - check_interval
 - SOFT
 - max_check_attempts
 - retry_interval



- /etc/icinga2/conf.d/myhosts.conf
- icinga2 daemon -C
- systemctl restart icinga2
- object
 - Host
 - Service
- import
 - templates
- address
- check_command
- vars

```
object Host "my-server1" {  
    address = "10.0.0.1"  
    check_command = "hostalive"  
}
```

```
object Service "ping4" {  
    host_name = "my-server1"  
    check_command = "ping4"  
}
```

```
object Service "http" {  
    host_name = "my-server1"  
    check_command = "http"  
}
```



Monitoring Agents

- Icinga Agent on Linux/Unix and Windows
- SSH on Linux/Unix
- SNMP on Linux/Unix and hardware
- SNMP Traps as passive check results
- REST API for passive external check results
- NSClient++ and WMI on Windows

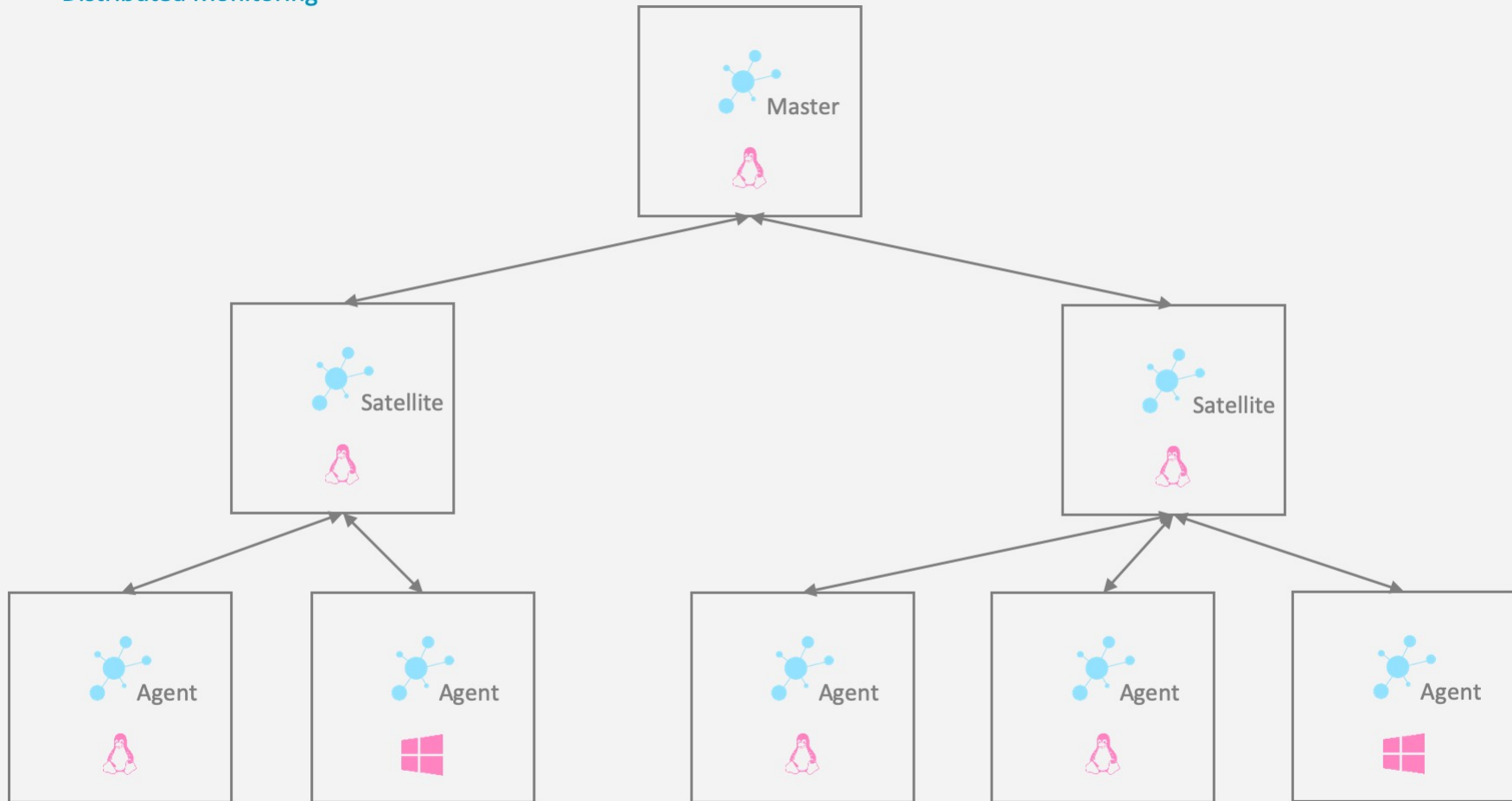


Distributed Monitoring

approved by
dsn.felk.cvut.cz

Roles

Distributed Monitoring





- master node
- client nodes
- 5665/TCP
- security
 - certs
 - tickets
- icinga2 node wizard
- `systemctl restart icinga2`



- icinga2 node wizard
- **n** – Satellite or Client setup
 - restart, fw 5665
- tickets
 - all nodes
 - icinga2 pki ticket --cn <client full name>



- install (Linux, Windows)
 - icinga2
 - monitoring-plugins
 - start, enable icinga2
- icinga2 node wizard
 - **y** - Satellite or Client setup
 - Master CN
 - **y** - establish a connection
 - ticket
 - **y** - accept config, commands
 - **y** - disable the inclusion of the conf.d
- restart, fw 5665



Configure Agent Monitoring

approved by
dsn.felk.cvut.cz

- master
- zones
- /etc/icinga2/zones.d/master/hosts.conf
- /etc/icinga2/zones.d/master/services.conf
- icinga2 daemon -C
- systemctl restart icinga2



Hosts

```
// for server01.example.com
object Endpoint "server01.example.com" {
}

object Zone "server01.example.com" {
  endpoints = [ "server01.example.com" ]
  parent = "master"
}

object Host "server01.example.com" {
  check_command = "hostalive" //check is executed
  on the master
  address = "192.168.43.63"

  vars.client_endpoint = name //follows the
  convention that host name == endpoint name
}
```

```
// for mydevapp.example.com
object Endpoint "mydevapp.example.com" {
}

object Zone "mydevapp.example.com" {
  endpoints = [ "mydevapp.example.com" ]
  parent = "master"
}

object Host "mydevapp.example.com" {
  check_command = "hostalive"
  address = "192.168.43.214"

  vars.client_endpoint = name
}
```



```
// Check System Load
apply Service "System Load" {
  check_command = "load"
  command_endpoint = host.vars.client_endpoint // Check executed
on client node
  assign where host.vars.client_endpoint
}

// Check number of running system Processes
apply Service "Process" {
  check_command = "procs"
  command_endpoint = host.vars.client_endpoint
  assign where host.vars.client_endpoint
}

// Check number of Logged in Users
apply Service "Users" {
  check_command = "users"
  command_endpoint = host.vars.client_endpoint
  assign where host.vars.client_endpoint
}
```

```
// Check System Disk Usage
apply Service "Disk" {
  check_command = "disk"
  command_endpoint = host.vars.client_endpoint
  assign where host.vars.client_endpoint
}

// Check for SWAP memory Usage
apply Service "SWAP" {
  check_command = "swap"
  command_endpoint = host.vars.client_endpoint
  assign where host.vars.client_endpoint
}

// SSH Service Check
apply Service "SSH Service" {
  check_command = "ssh"
  command_endpoint = host.vars.client_endpoint
  assign where host.vars.client_endpoint
}

apply Service "Ping" {
  check_command = "ping4"
  assign where host.address
}
```



...